BOUT++ overview

Ben Dudson and the BOUT++ team

BOUT++ workshop, LLNL
14th August 2018

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Acknowledgements

Many people have contributed to BOUT++

30 contributors to the github version
17 since version 3.0 (Oct 2016), with 3813 commits (~ 5.8 per day)

Particular thanks to:

Peter Hill, David Dickinson, Joseph Parker, David Schworer, John Omotani, Michael Loiten, Jens Madsen, Jarrod Leddy, Nick Walkden, Adam Dempsey, Haruki Seto, Brendan Shanahan, Erik Grinaker, Xiang Liu, Maxim Umansky, Matt Thomas

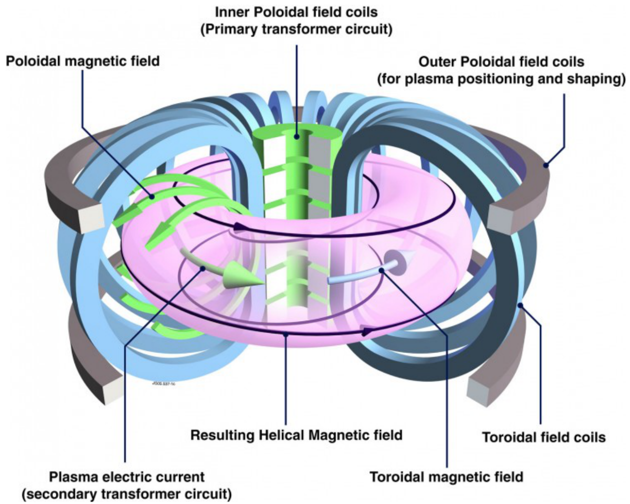
Outline

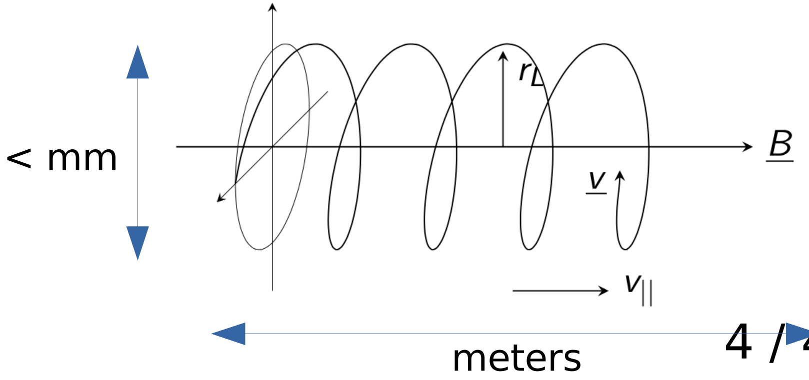
- Plasma modelling, drift-reduced models
- BOUT++ code structure
- Recent changes (last ~ 2-3 years)
  - Electromagnetic field solvers, global modes
  - Complex magnetic geometries (FCI scheme)
- Getting started
- Contributing to BOUT++

Tokamak plasmas

Typical parameters

- Pressure ~ 1 atmosphere
- Temperature ~ 100 million °C
- Density ~ 10^{20} m^{-3}
  (3.3 x 10^{-7} kg/m^3 . Air ~ 1.2 kg/m^3)
- Sound speed ~ 10^5 – 10^6 m/s
- Mean free paths ~ 1 – 100 m
- Ion gyrofrequency ~ GHz
- Global evolution timescale ~ 10s of ms

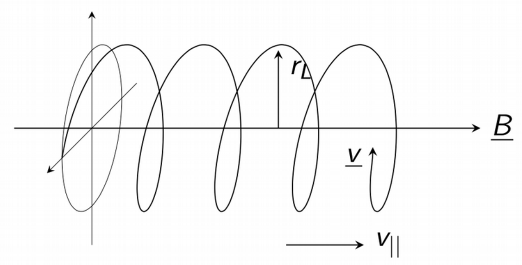




Example model

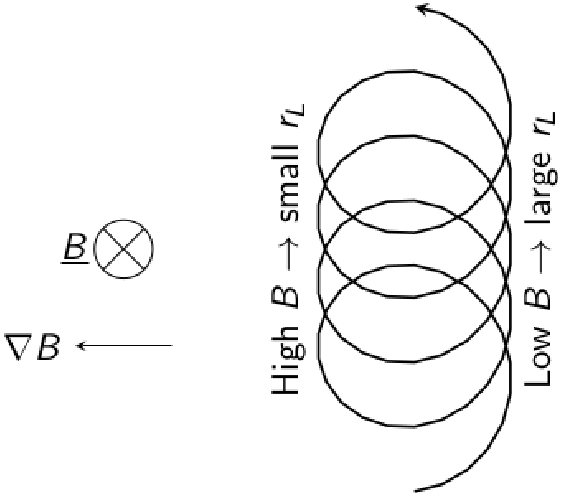
A simple model can be derived from continuity of particles and charge:

$$\frac{\partial n}{\partial t} + \nabla \cdot \left(\underline{v} n\right) = 0$$
$$\nabla \cdot \underline{J} = 0$$



Particles drift due to electric fields, magnetic inhomogeneities and time-varying fields

$$\underline{v} = \underline{b} v_{||} + \frac{\underline{E}\times\underline{B}}{B^2} + \underbrace{\frac{T}{B}\frac{\underline{B}\times\nabla B}{B^2}}_{\text{Ignore for ions } (T_i = 0)} + \underbrace{\frac{m}{qB^2}\frac{d\underline{E}_\perp}{dt}}_{\text{Ignore for electrons } (m_e \text{ small})}$$



Example model

Substituting this velocity into density and current equations gives:

$$\frac{\partial n}{\partial t} = \boxed{-\nabla \cdot \left(n\underline{b}v_{||e}\right)} \boxed{- \nabla \cdot \left(n\frac{\underline{E} \times \underline{B}}{B^2}\right)} \boxed{- \nabla \cdot \left(\frac{nT}{B}\underbrace{\frac{\underline{B} \times \nabla B}{B^2}}_{\sim 2/R_c}\right)}$$

$$\boxed{\nabla \cdot \left(\frac{m_i n}{eB^2}\frac{d\underline{E}_\perp}{dt}\right)} = -\nabla \cdot J_{||} - \nabla \cdot \left(\frac{-enT}{B}\frac{\underline{B} \times \nabla B}{B^2}\right)$$

Assuming incompressible, neglecting ion parallel flow, and making thin layer (Boussinesq) approximation gives:

$$\frac{\partial n}{\partial t} = \boxed{-\frac{1}{B}\underline{b} \times \nabla\phi \cdot \nabla n} \boxed{+ \frac{1}{e}\nabla \cdot J_{||}} \boxed{+ \frac{2T_e}{BR_c}\frac{\partial n}{\partial z}}$$

$$\boxed{\begin{aligned} \frac{\partial \omega}{\partial t} &= -\frac{1}{B}\underline{b} \times \nabla\phi \cdot \nabla\omega \\ \omega &= \frac{m_i n_0}{B^2}\nabla_\perp^2\phi \end{aligned}} + \nabla \cdot J_{||} + \frac{2eT_e}{BR_c}\frac{\partial n}{\partial z}$$

Example model

$$\frac{\partial n}{\partial t} = -\frac{1}{B}\underline{b}\times\nabla\phi\cdot\nabla n + \frac{1}{e}\nabla\cdot J_{||} + \frac{2T_e}{BR_c}\frac{\partial n}{\partial z}$$

$$\frac{\partial \omega}{\partial t} = \boxed{-\frac{1}{B}\underline{b}\times\nabla\phi\cdot\nabla\omega} + \nabla\cdot J_{||} + \frac{2eT_e}{BR_c}\frac{\partial n}{\partial z}$$

Incompressible fluid e.g.
Arakawa '65 and extensions

Gradients along
magnetic field
(compressible fluid)

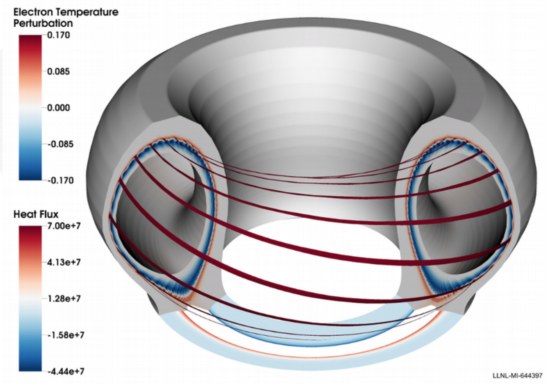
$$\omega = \frac{m_i n_0}{B^2}\nabla_\perp^2\phi$$

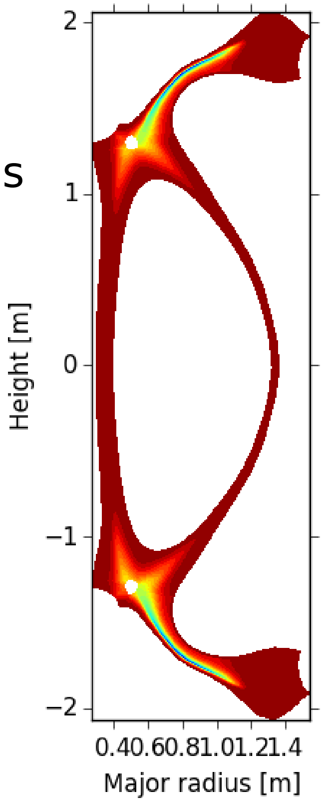
Need to invert Laplacian to evaluate electric potential.
→ Spectral methods, multigrid schemes

Peterson, Hammett:
http://dx.doi.org/10.1137/120888053

BOUT++

- An open-source library for solving PDEs in curvilinear geometry
- Specialised operators for plasma applications
- Finite difference / volume code in 3D mapped multi-block geometry
- Solves nonlinearly coupled hyperbolic, parabolic and elliptic equations
- MPI parallelised, scales to ~ 4,000 cores (depending on problem)
- Turbulence ~10^6-10^8 unknowns, ~10^5 core-hours





BOUT++: example model

https://github.com/boutproject/BOUT-dev/tree/master/examples/blob2d

$$\frac{\partial n}{\partial t} = -\frac{1}{B}\underline{b} \times \nabla\phi \cdot \nabla n + \frac{1}{e}\nabla \cdot J_{||} + \frac{2T_e}{BR_c}\frac{\partial n}{\partial z}$$

$$\frac{\partial \omega}{\partial t} = -\frac{1}{B}\underline{b} \times \nabla\phi \cdot \nabla \omega + \nabla \cdot J_{||} + \frac{2eT_e}{BR_c}\frac{\partial n}{\partial z}$$

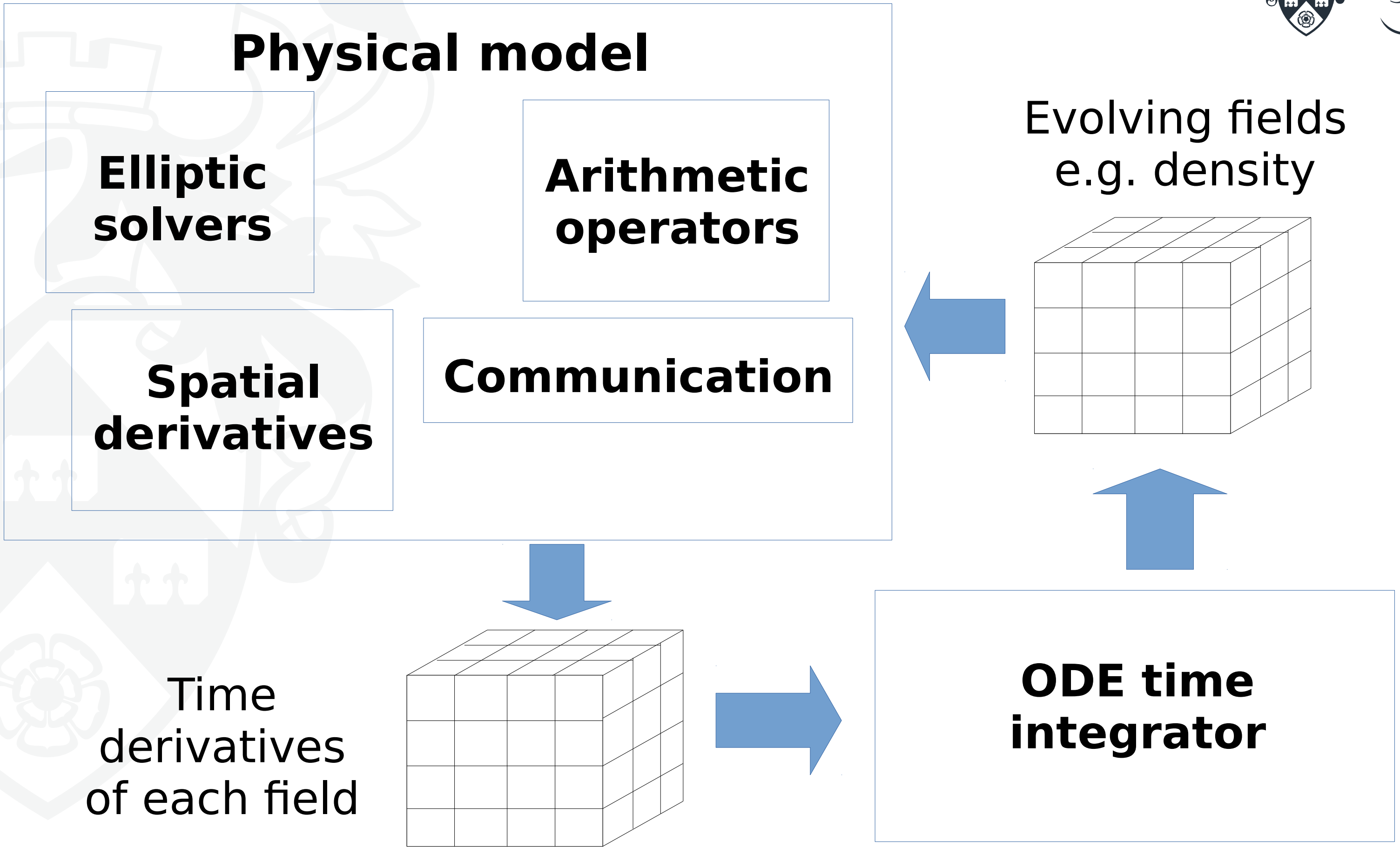
$$\omega = \frac{m_i n_0}{B^2}\nabla_\perp^2 \phi$$

```
ddt(n) = -vE_Grad(n,phi+phi0)
         + Div_par(Jpar)
         + 2*DDZ(n)*(rho_s/R_c);

ddt(omega) = -bracket(phi,omega)
             + Div_par(Jpar)
             + 2*DDZ(n)*(rho_s/R_c);

phi = phiSolver->solve(omega);
```

Outline

- Plasma modelling, drift-reduced models
- BOUT++ code structure
- Recent changes (last ~ 2-3 years)
  - Electromagnetic field solvers, global modes
  - Complex magnetic geometries (FCI scheme)
- Getting started
- Contributing to BOUT++

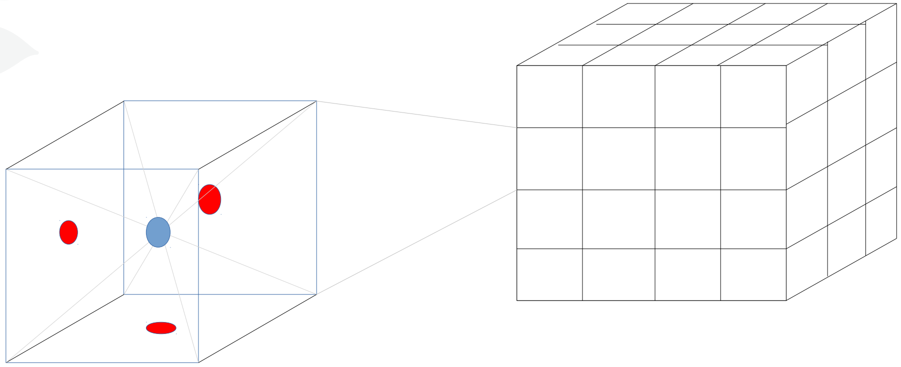
BOUT++: Method of Lines



BOUT++ mesh

3D mesh, locally logically Cartesian

Quantities usually cell centre; some staggered cell face values



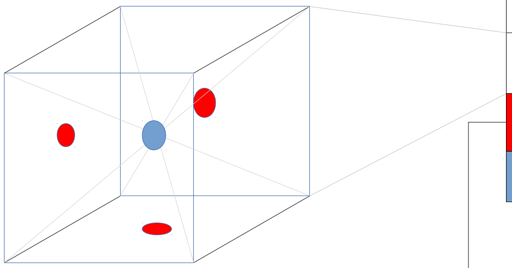
Represented as Field3D objects

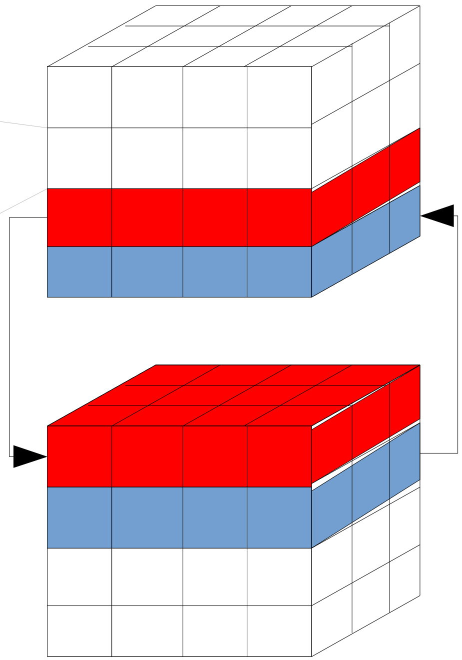
```
Field3D n, T;
…
Field3D p = n * T;
```

BOUT++ mesh

3D mesh, locally logically Cartesian

Guard/ghost cells used for boundaries and communication





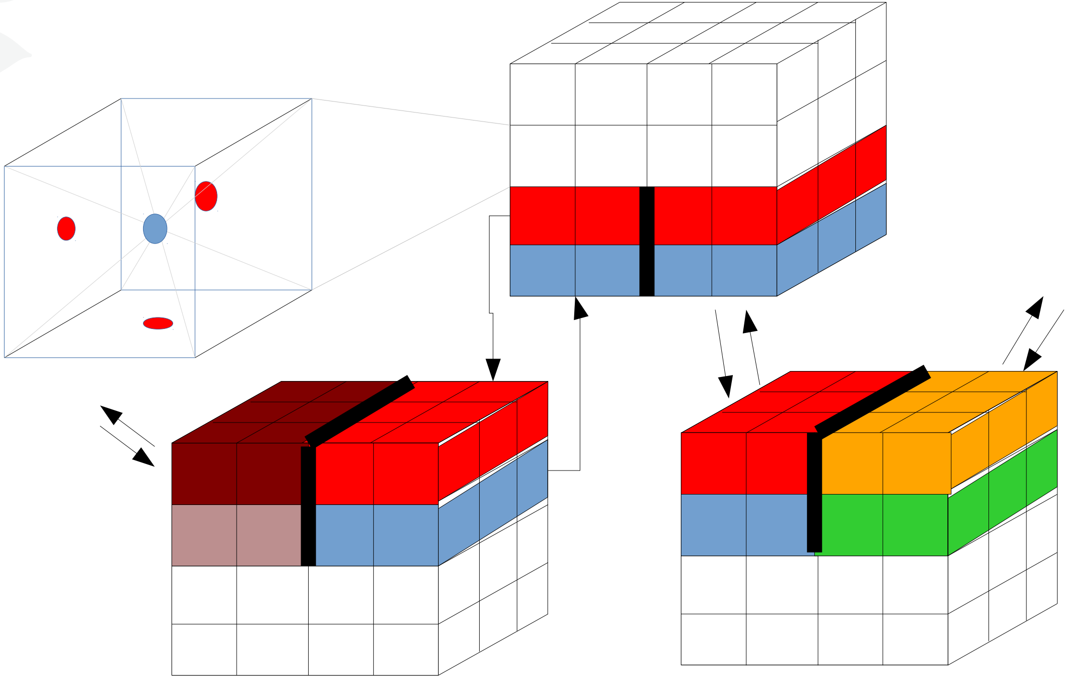
Communication handled by Mesh:

```
mesh->communicate(n, T);
```

BOUT++ mesh

3D mesh, locally logically Cartesian

More complicated topologies handled using branch cuts

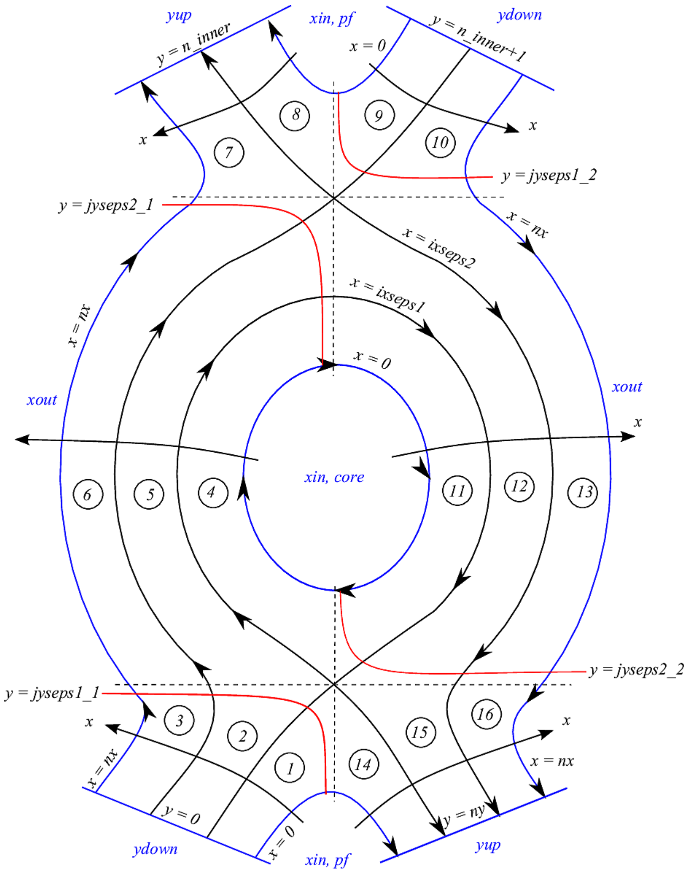


BOUT++ mesh

3D mesh, locally logically Cartesian

More complicated topologies handled using branch cuts

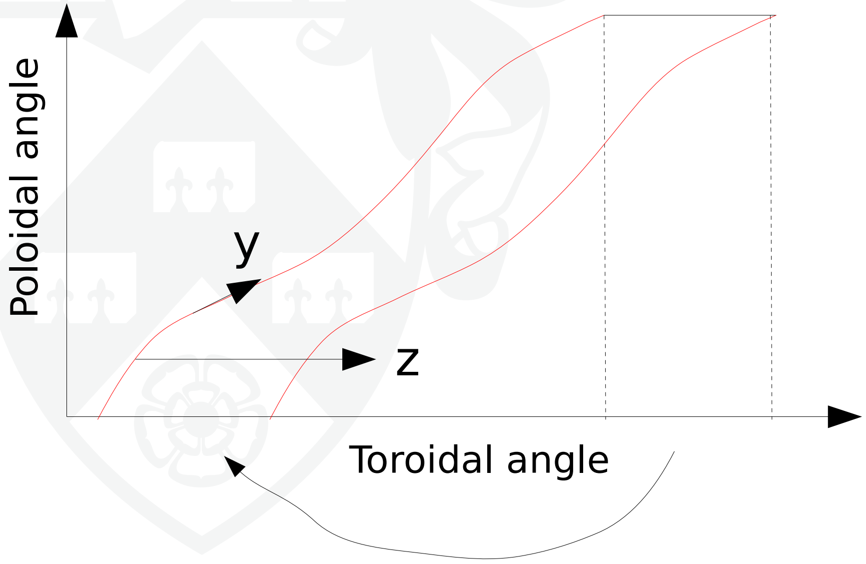
Tokamak edge simulations usually have at least one “X-point” and a hole in the domain.



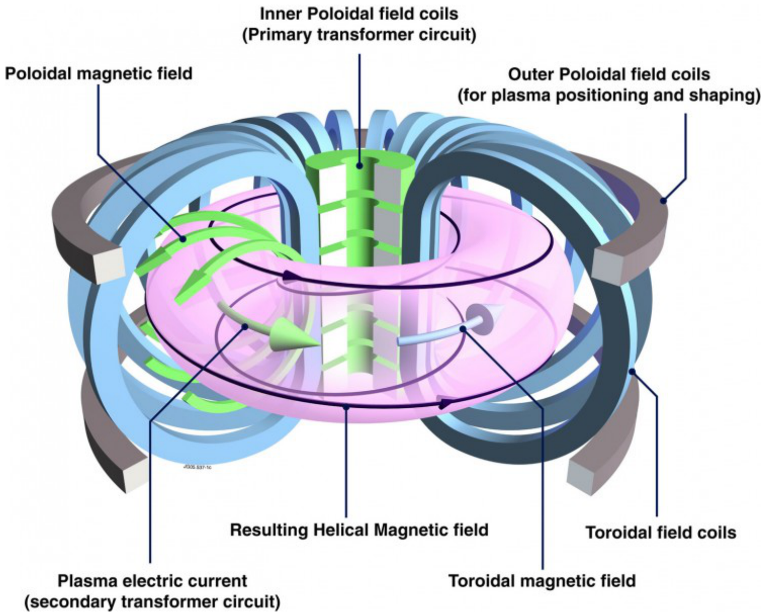
Coordinates

Usually curved, usually non-orthogonal aligned to the helical magnetic field.

In the core of the plasma the domain is a twisted ribbon



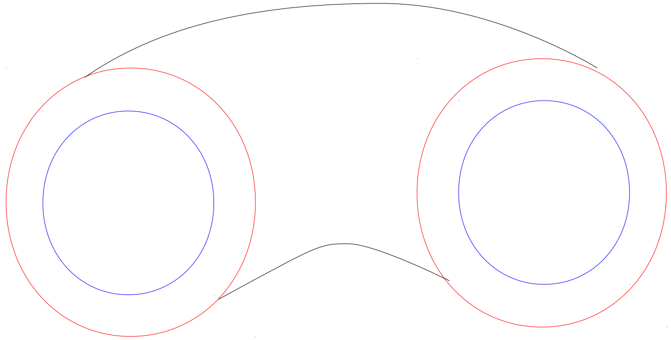
Twist-shift boundary (matching) condition

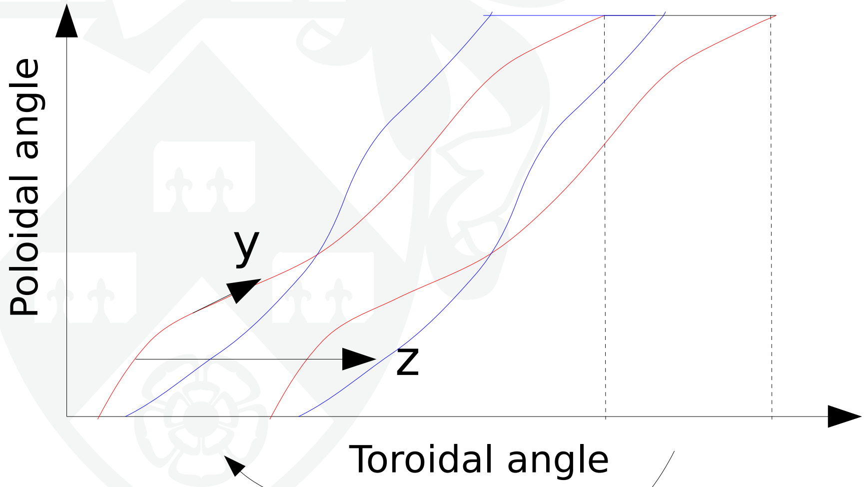


Coordinates

Usually curved, usually non-orthogonal aligned to the helical magnetic field.

In the core of the plasma the domain is a twisted ribbon





Twist-shift boundary (matching) condition

Pitch of the magnetic field varies

Leads to shearing of coordinate system

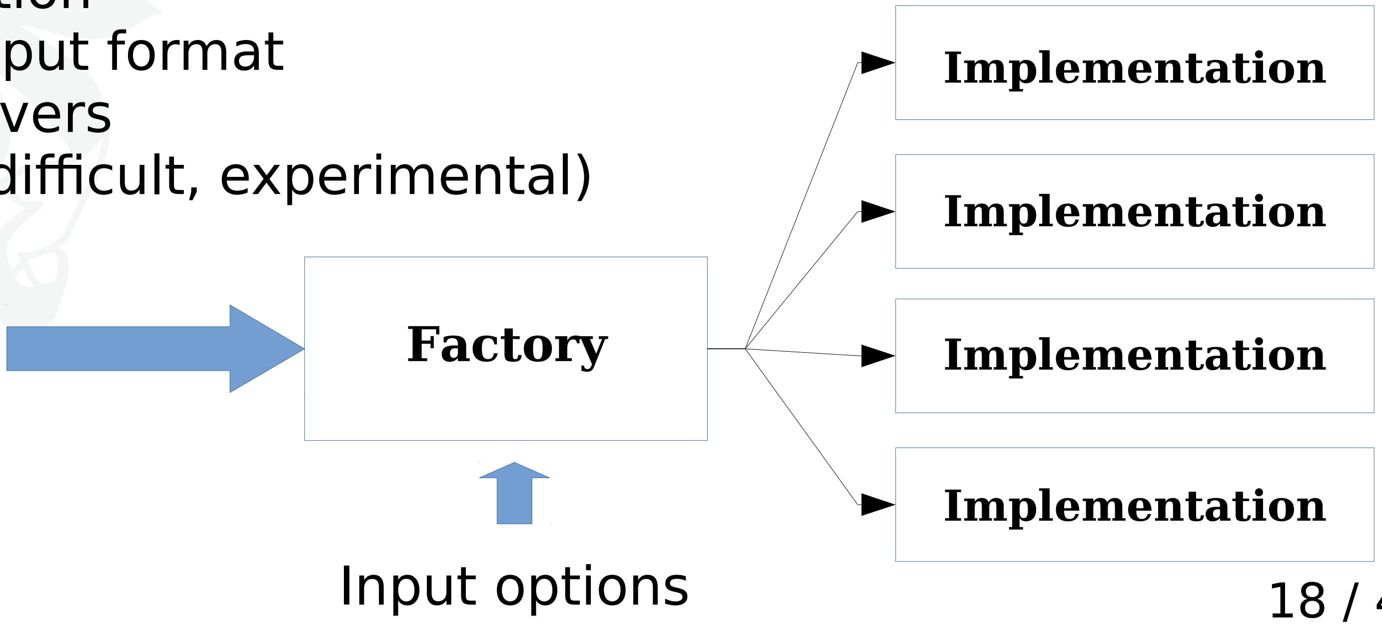
Often necessary to re-map onto local coordinates to reduce shearing

Shifts in toroidal angle → FFTs

Modular structure

Many parts of BOUT++ have multiple implementations.
Changed with an input option, with no/minimal user code changes.

- Time integration
- File input/output format
- Laplacian solvers
- Mesh (more difficult, experimental)



Outline

- Plasma modelling, drift-reduced models
- BOUT++ code structure
- Recent changes (last ~ 2-3 years)
  - Electromagnetic field solvers, global modes
  - Complex magnetic geometries (FCI scheme)
- Getting started
- Contributing to BOUT++

Solving potentials

Need to invert a Laplacian to obtain potential from vorticity:

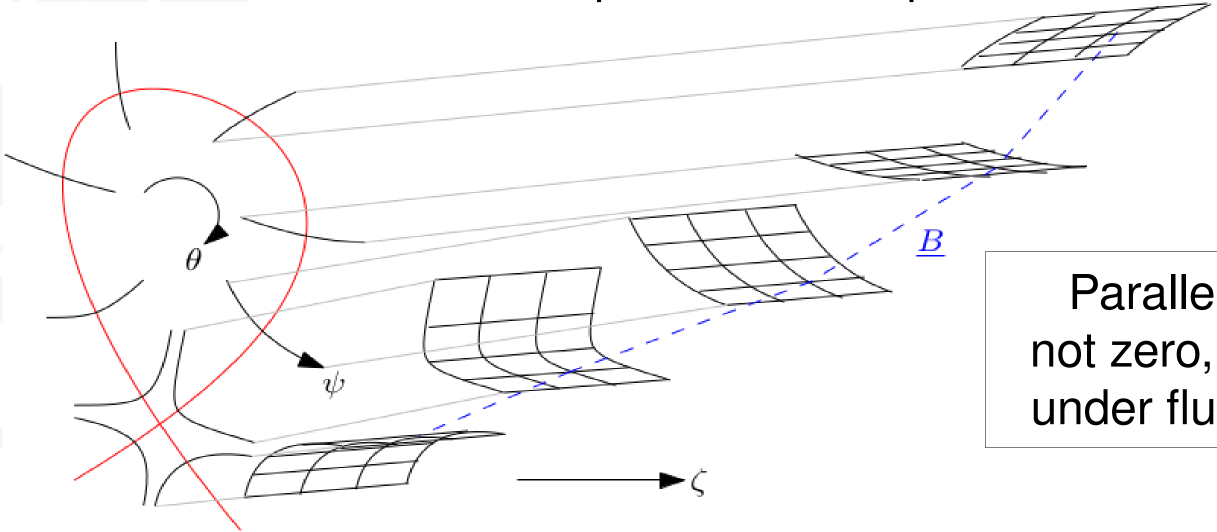
$$\nabla\cdot\left(\frac{m_i n}{B^2}\nabla_\perp\phi\right) = \frac{1}{J}\frac{\partial}{\partial u^i}\left(J\frac{m_i n}{B^2}g^{ij}\left(\nabla_\perp\phi\right)_j\right)$$

Coordinates

$$x = \psi \qquad y = \theta$$

$$z = \zeta - \int_{\theta_0}^{\theta}\frac{B_\zeta h_\theta}{B_\theta R}d\theta$$

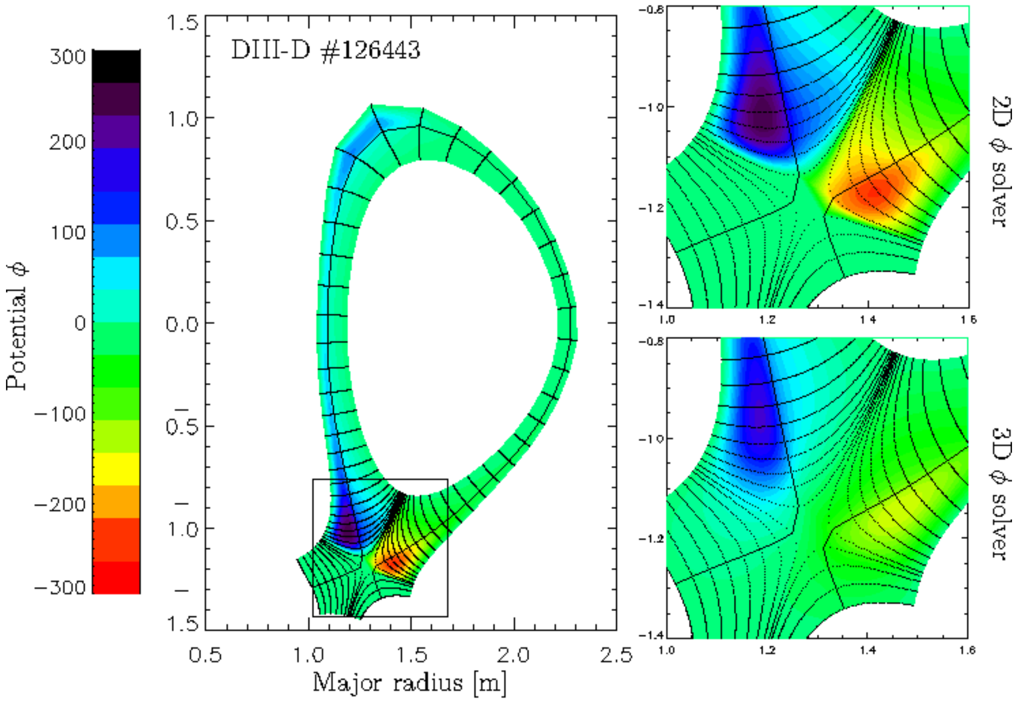
Field-aligned coordinates uses toroidal planes as drift planes:



Parallel component not zero, but neglected under flute assumption

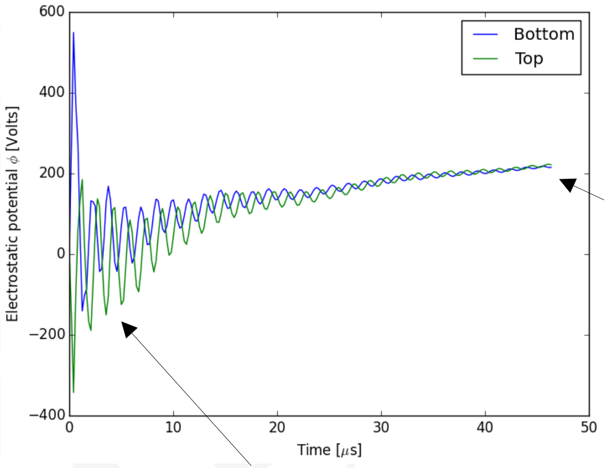
Solving n=0 modes

- Solving on toroidal planes results in large savings: Each X-Z slice can be solved separately
- For low n modes this simplification breaks down. Results in unphysical electric fields close to the X-point
- A 2D solver has been developed for the n=0 mode using PETSc



B D Dudson, J Leddy, "Hermes", PPCF 59(5), 054010 (2017)

Solving n=0 modes

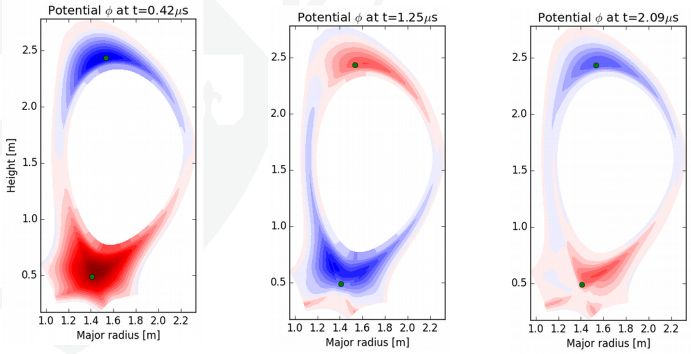


Drifts, currents

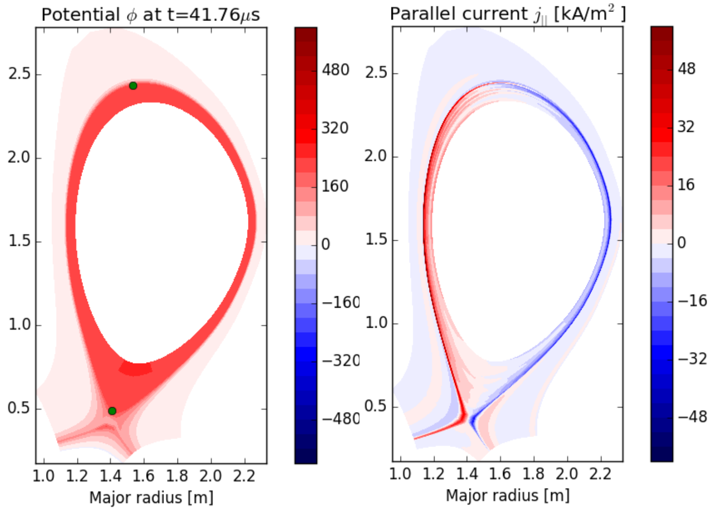
Evolve n=0 electric field

Transients and steady state

Transient oscillation



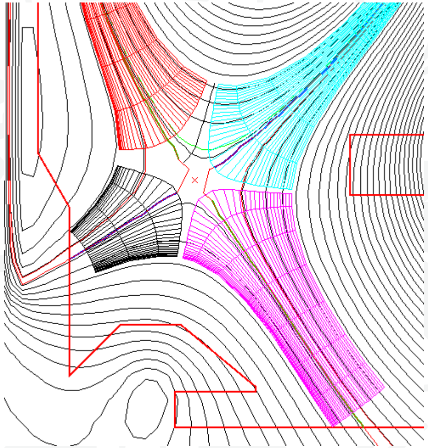
Steady state



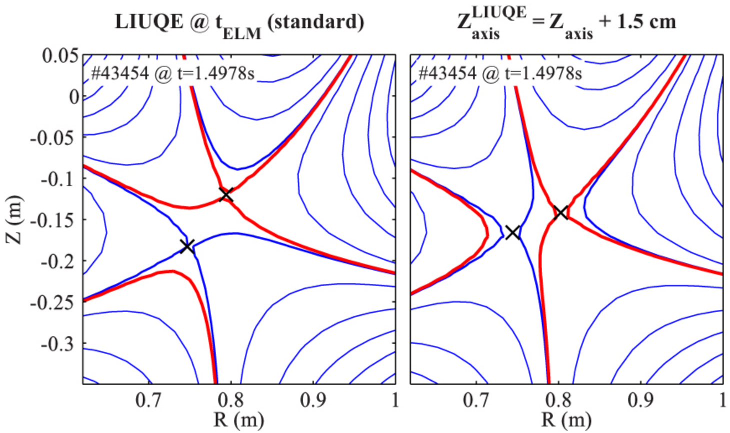
B D Dudson, J Leddy, “Hermes”, PPCF 59(5), 054010 (2017)

Complicated magnetic fields

X-point

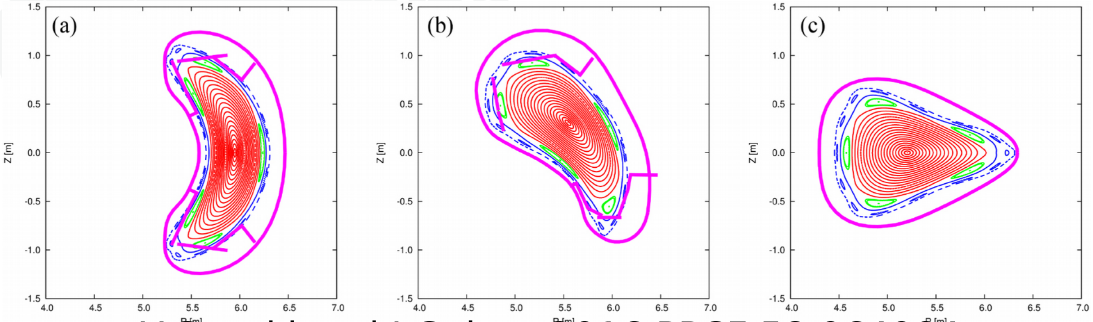


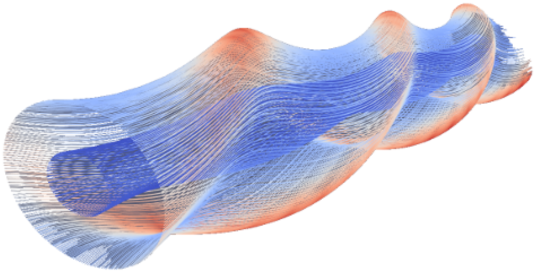
Snowflake



W.A.J. Vijvers et al 2014 Nucl. Fusion 54 023009

Stellarator



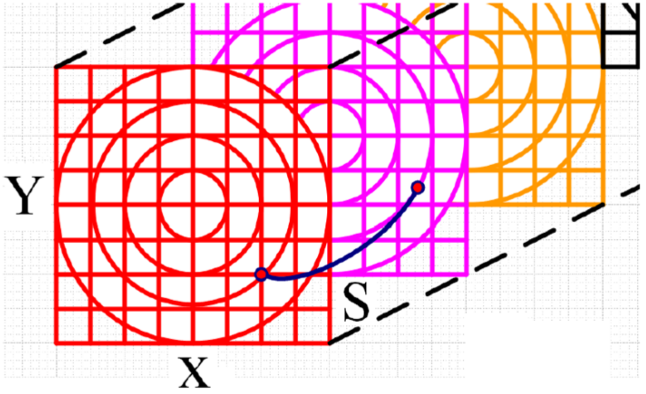


Y Suzuki and J Geiger 2016 PPCF 58 064004

Flux coordinate independent

Field aligned grids are required in order to:
→ Significantly reduce resolution in one dimension
→ Accurately capture anisotropic transport

Field-aligned coordinates have singularities
→ Reduces convergence rate
→ Can lead to numerical instability
→ Difficult to resolve features close to the X-point



The FCI approach is to use:
- Well-behaved local coordinates,
- field-line following and interpolation to calculate parallel derivatives

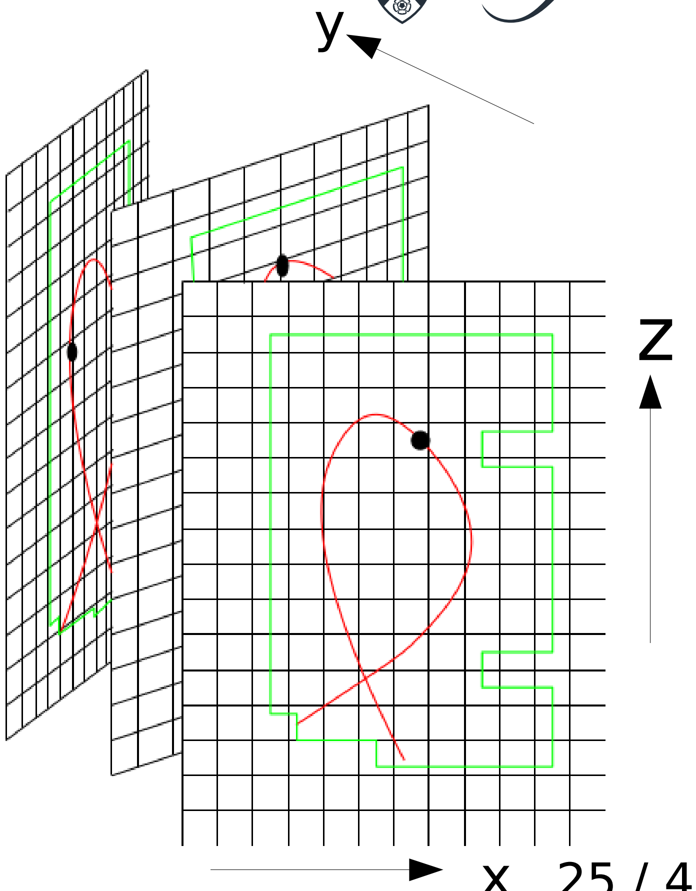
[1] F Hariri and M Ottaviani CPC **184** 2419 (2013)
[2] B Shanahan, P Hill and B Dudson. Journal of Physics; Conference Series **775**, 012012 (2016)
[3] A.Stegmeir et al. CPC 198, 139-153 (2016)
[4] P Hill, B Shanahan and B Dudson CPC **213**, 9-18 (2017)

Tokamak simulations

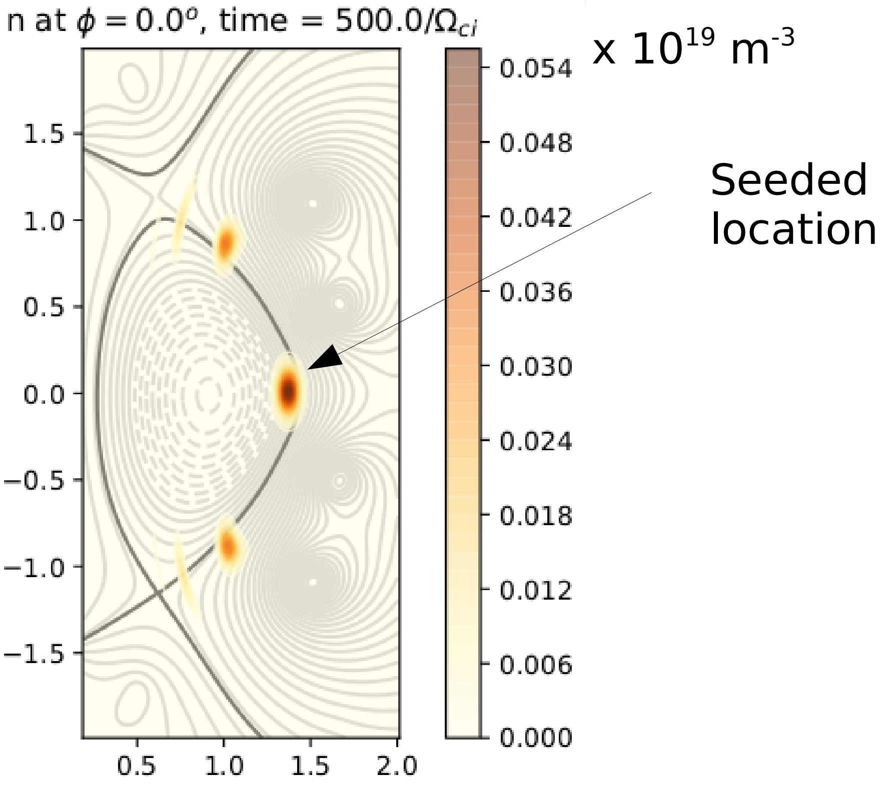
MAST geometry (shot 14220)
R = 0.2 → 2m ; Z = -1.5 → +1.5m

Simple sheath boundary conditions at walls: $v_{||} = cs$

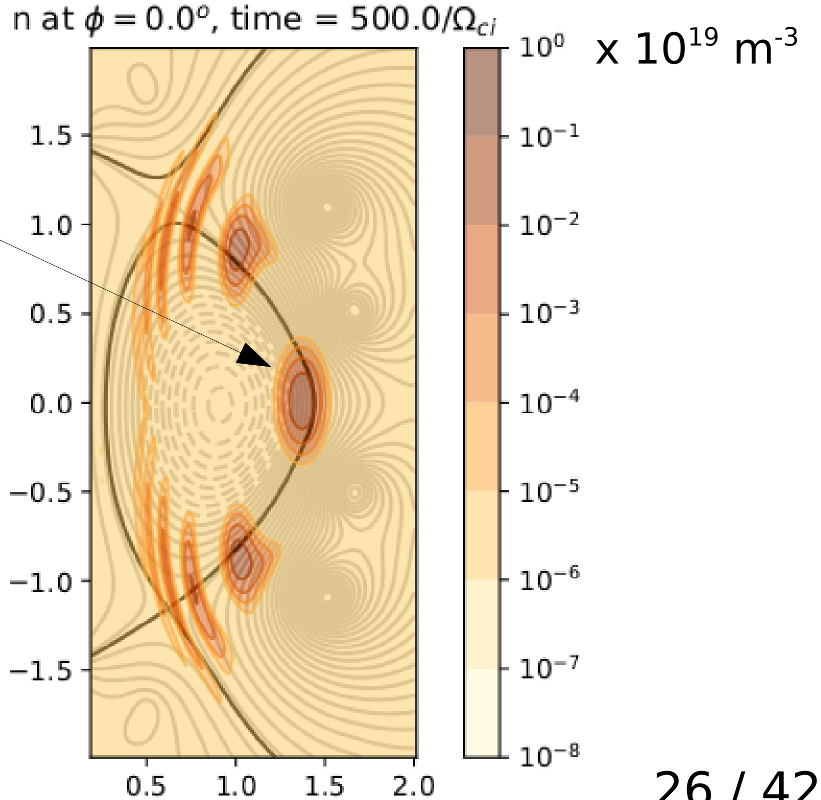
Electromagnetic field evolved (Alfven waves), but B perturbation **not** included (for now)



Plasma density

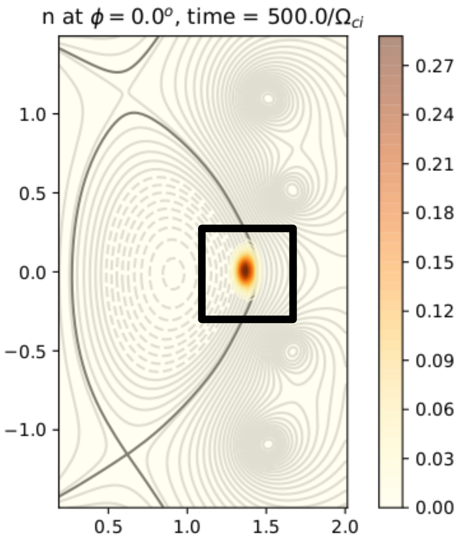


Plasma density (log)

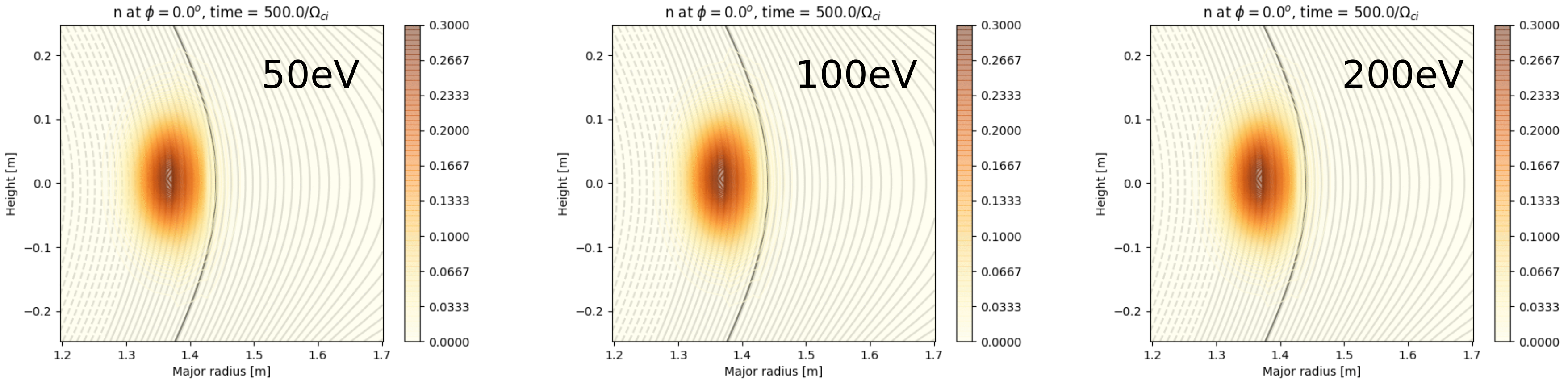


Tokamak filaments

(Under-resolved proof-of-principle)



- Start with an isolated plasma “ball” (no background)
- Evolve only parallel diffusion (10^5 m/s) for ~10μs → Filament
- Turn on drifts, electric fields



Stellarator simulations

- New grid generator for (nearly) arbitrary shapes
- Initial blob studies in “straight stellarators”
- Simulations of Wendelstein 7-X started

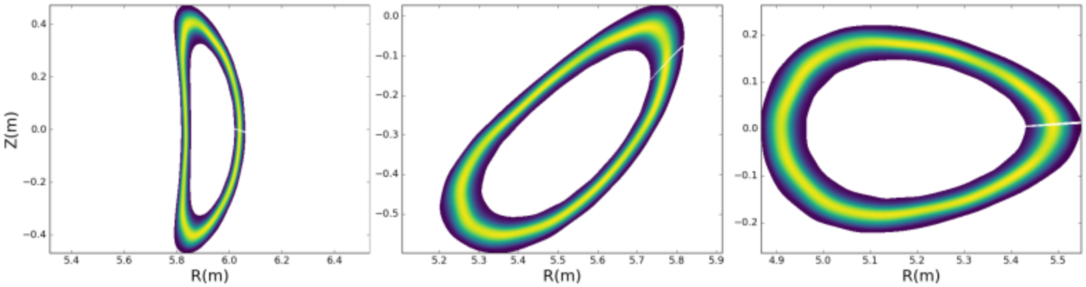
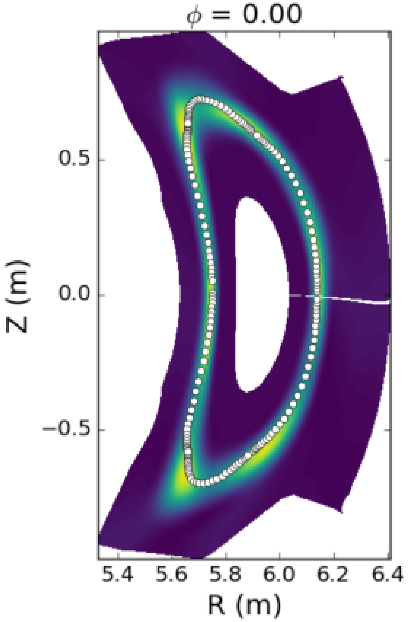


FIG. 14: Three cross sections of the Wendelstein 7-X stellarator indicating flux surfaces as traced by a parallel heat diffusion equation in BSTING



(b) Simulated surfaces for the bean-shaped-cross section

B Shanahan, B Dudson, P Hill, “BSTING”, Submitted to PPCF (2018)

Other recent changes

Some of the many fixes and new features

1. Ability to handle complex 3D magnetic field structures
2. Better electromagnetic field solvers
3. Running BOUT++ from Python
4. Faster, clearer loops, C++11 features**
5. Better differential operators, flux-conserving schemes
6. More usable, easier to install, debug, optimise, extend

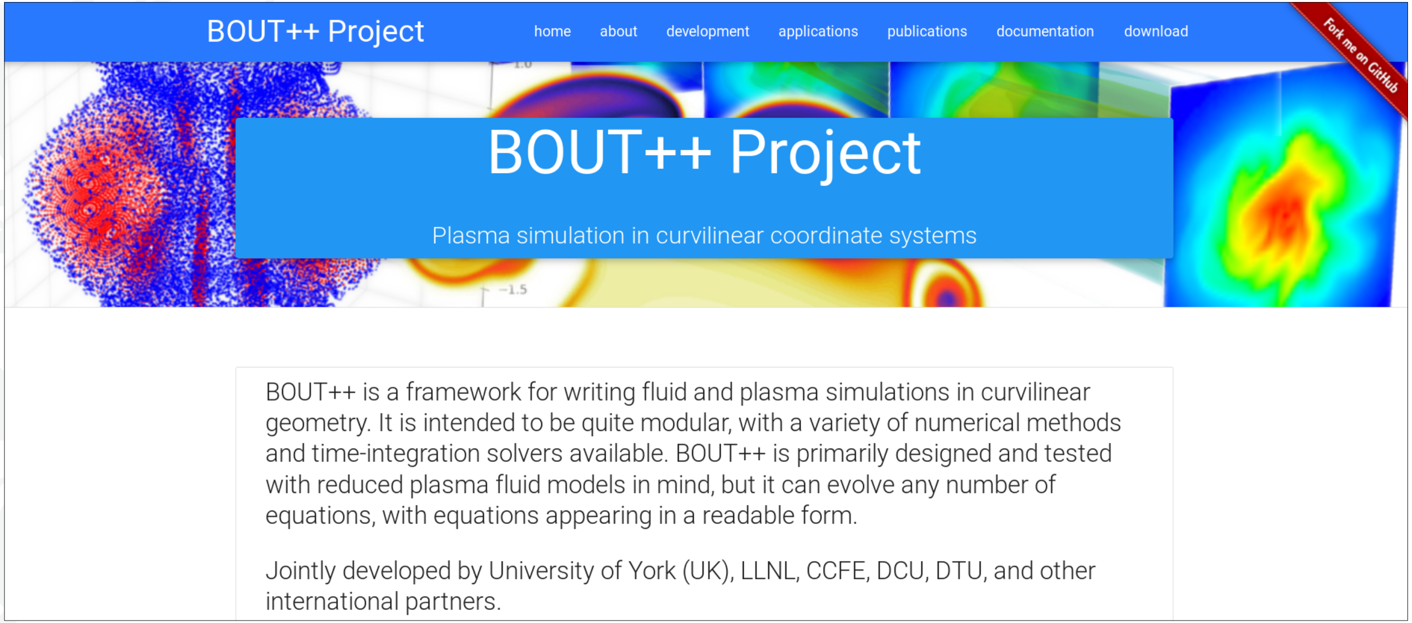
** See performance talk (tomorrow)

Outline

- Plasma modelling, drift-reduced models
- BOUT++ code structure
- Recent changes (last ~ 2-3 years)
  - Electromagnetic field solvers, global modes
  - Complex magnetic geometries (FCI scheme)
- Getting started
- Contributing to BOUT++

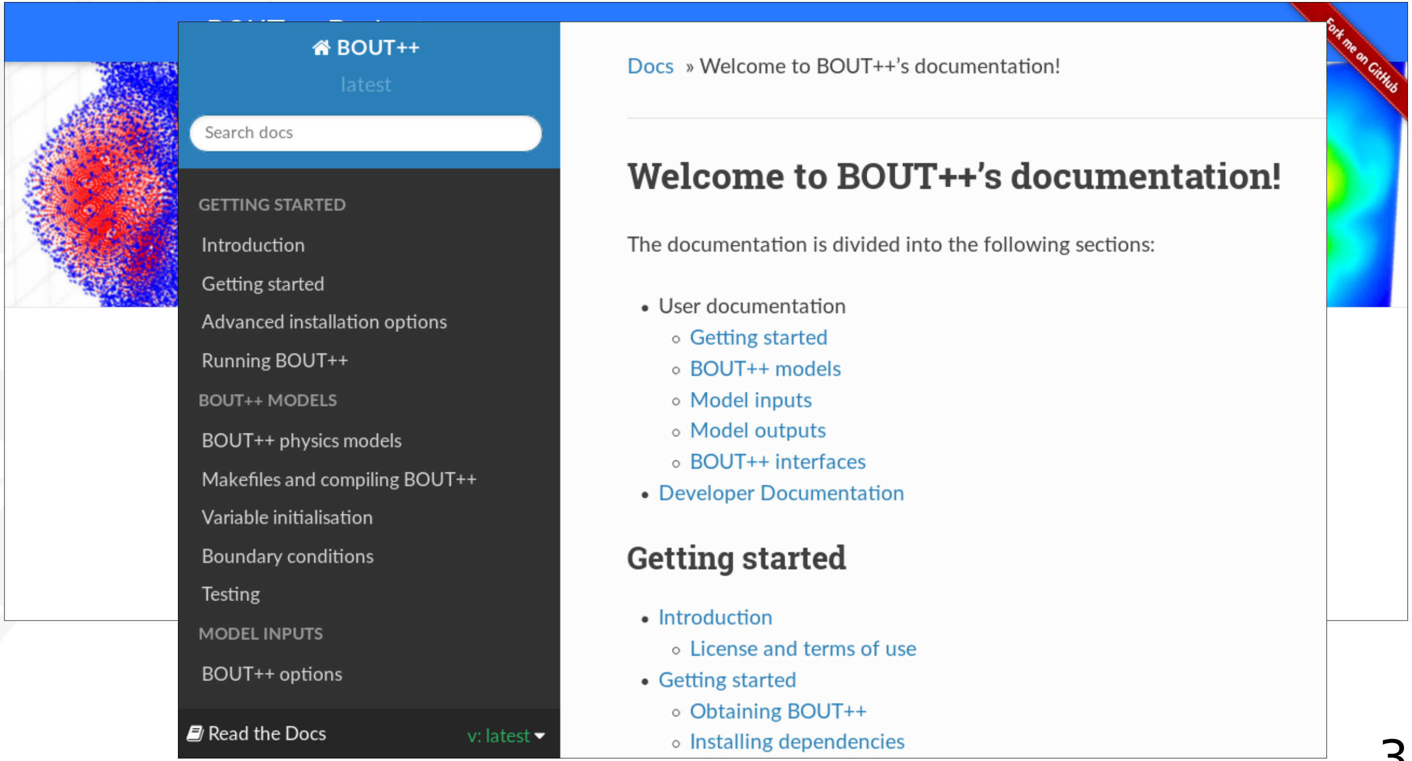
Getting started

http://boutproject.github.io/



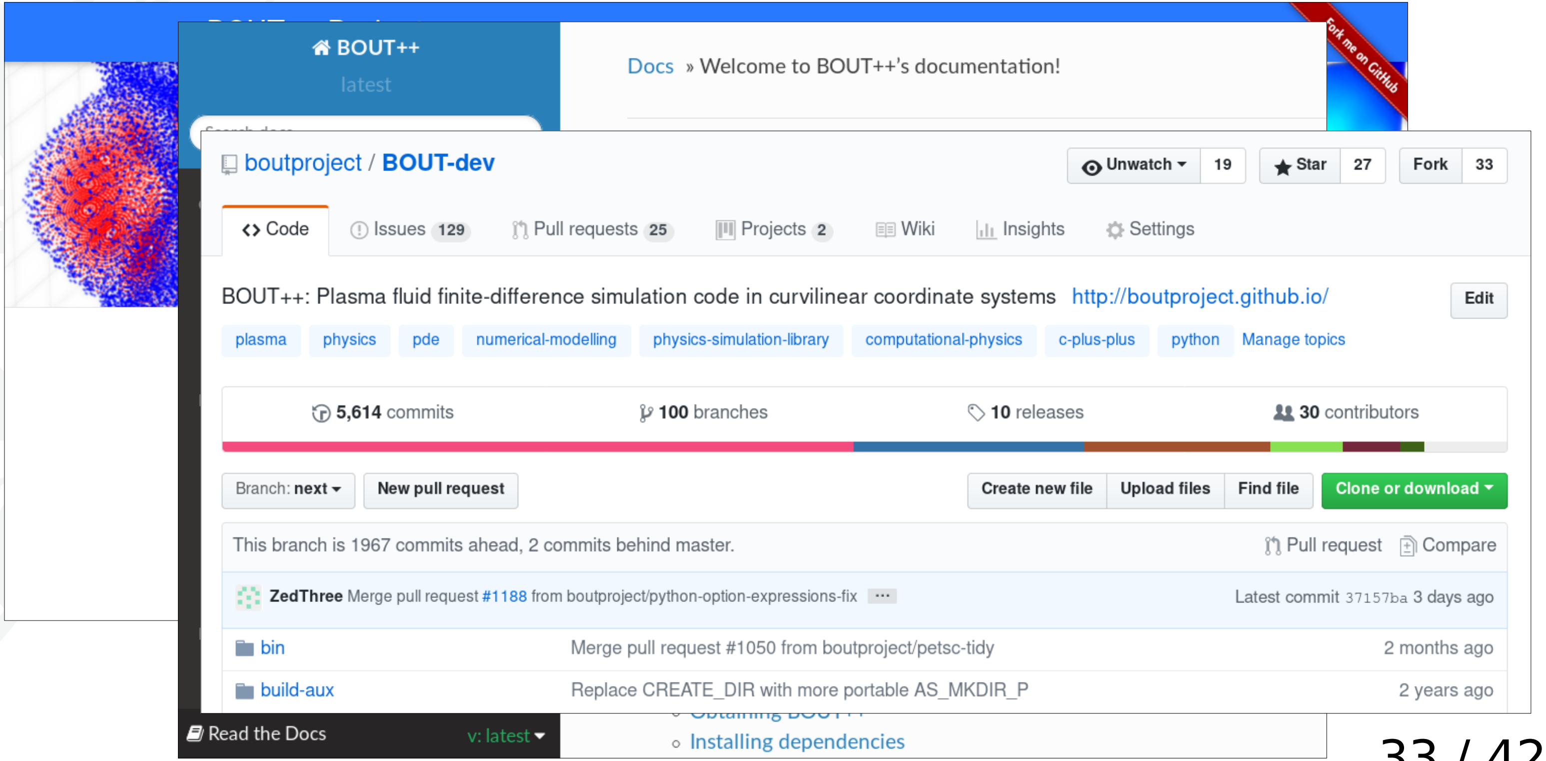
Getting started

http://bout-dev.readthedocs.io



Getting started

https://github.com/boutproject/BOUT-dev



Install docker

Jarrod Leddy (Tech-X)

- Provides a reproducible environment (like a virtual machine or BSD jail)

Ubuntu linux
https://docs.docker.com/install/linux/docker-ce/ubuntu/

Mac:
https://docs.docker.com/docker-for-mac/install/

Windows:
https://docs.docker.com/docker-for-windows/install/

Others:
https://docs.docker.com/v17.12/install/

BOUT++ docker image

Jarrod Leddy (Tech-X)

- Provides a reproducible environment (like a virtual machine or BSD jail)
- Preinstalled with PETSc, SLEPC, SUNDIALS, BOUT++, editors and python analysis tools

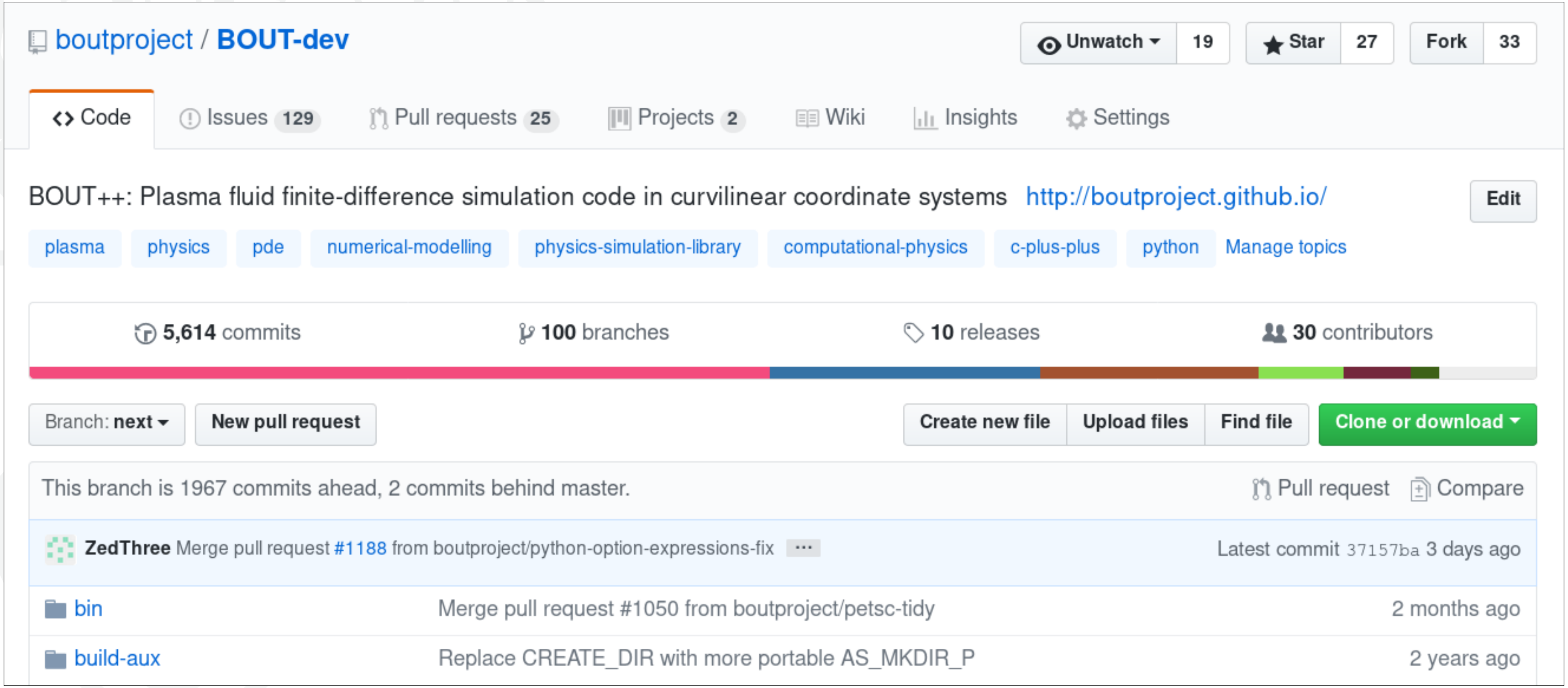
https://hub.docker.com/r/boutproject/bout-next/

```
$ sudo docker pull boutproject/boutproject/bout-next:9f4c663-petsc
$ sudo docker run --rm -it boutproject/bout-next:9f4c663-petsc
```

Outline

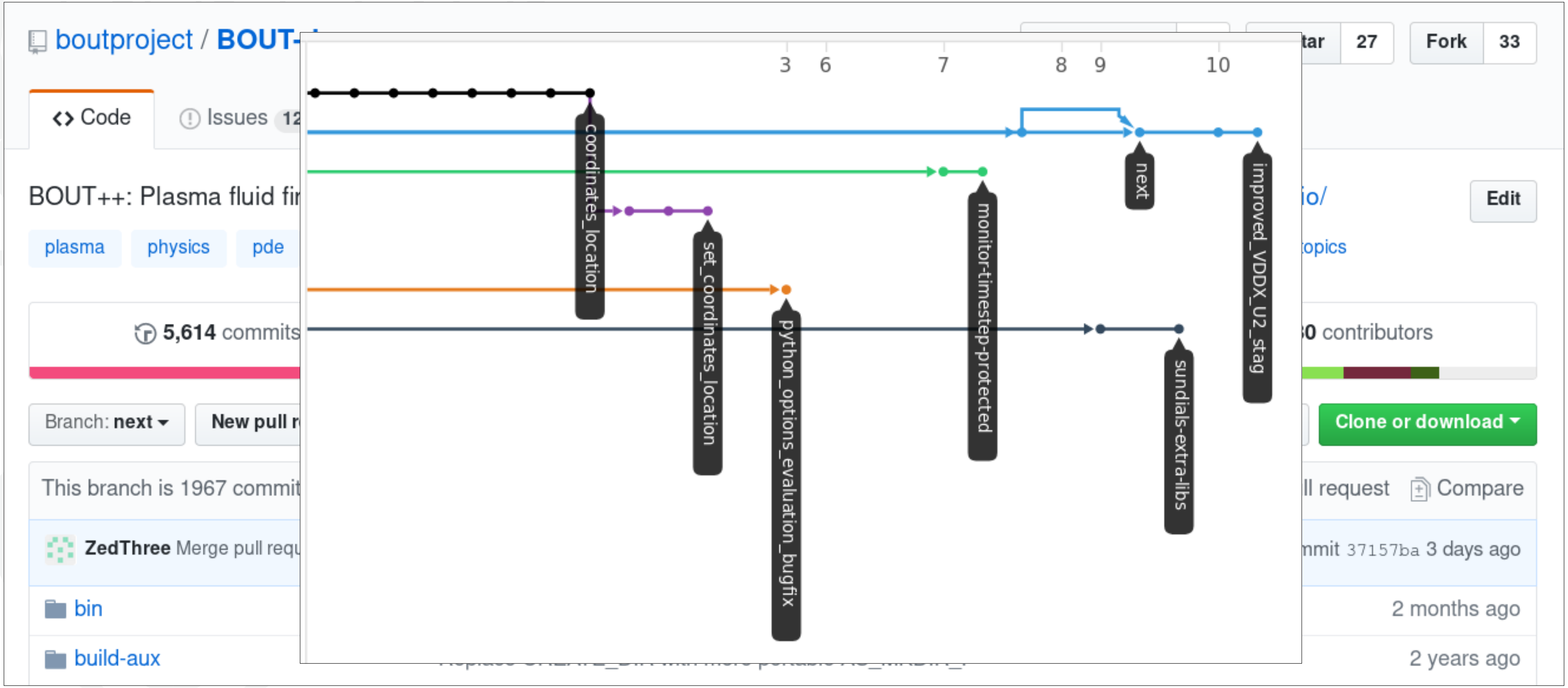
- Plasma modelling, drift-reduced models
- BOUT++ code structure
- Recent changes (last ~ 2-3 years)
  - Electromagnetic field solvers, global modes
  - Complex magnetic geometries (FCI scheme)
- Getting started
- Contributing to BOUT++

Development process



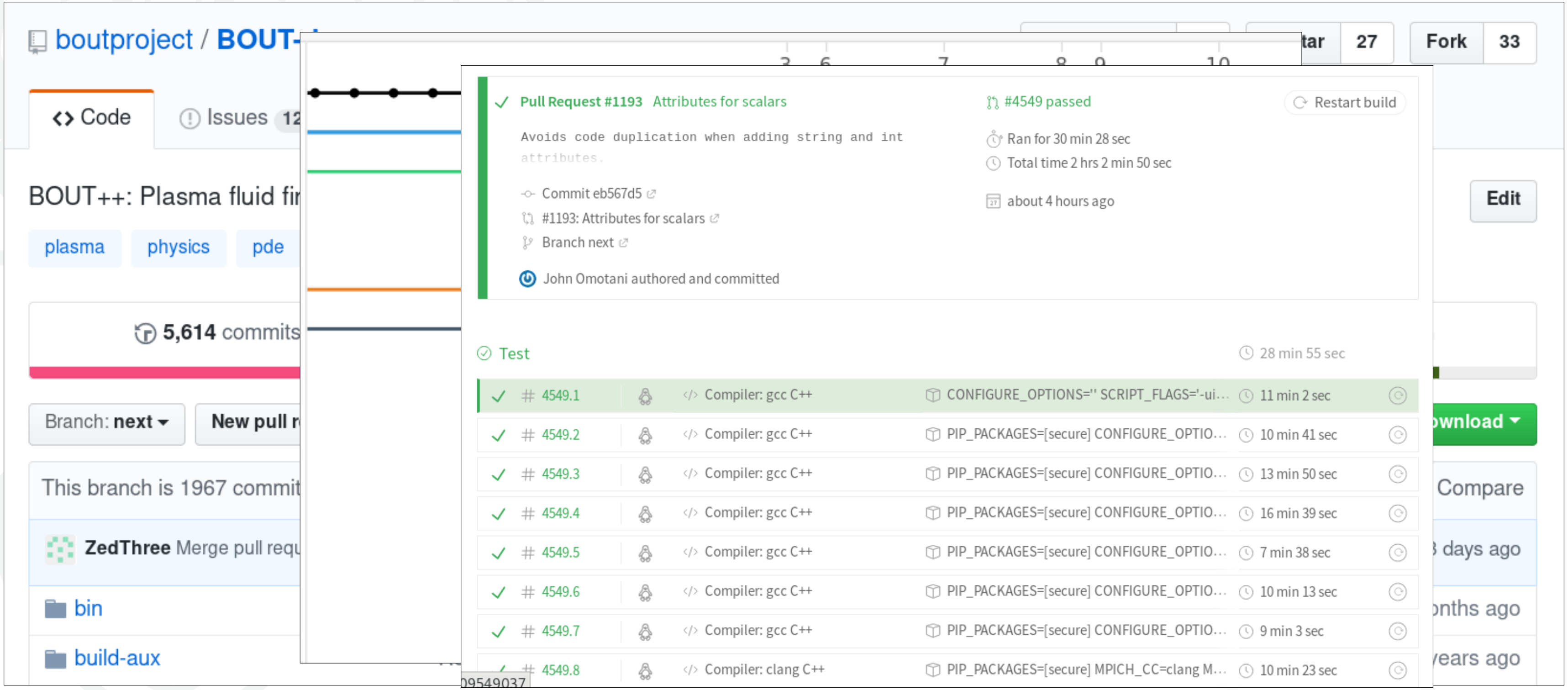
Code hosted on **Github**

Development process



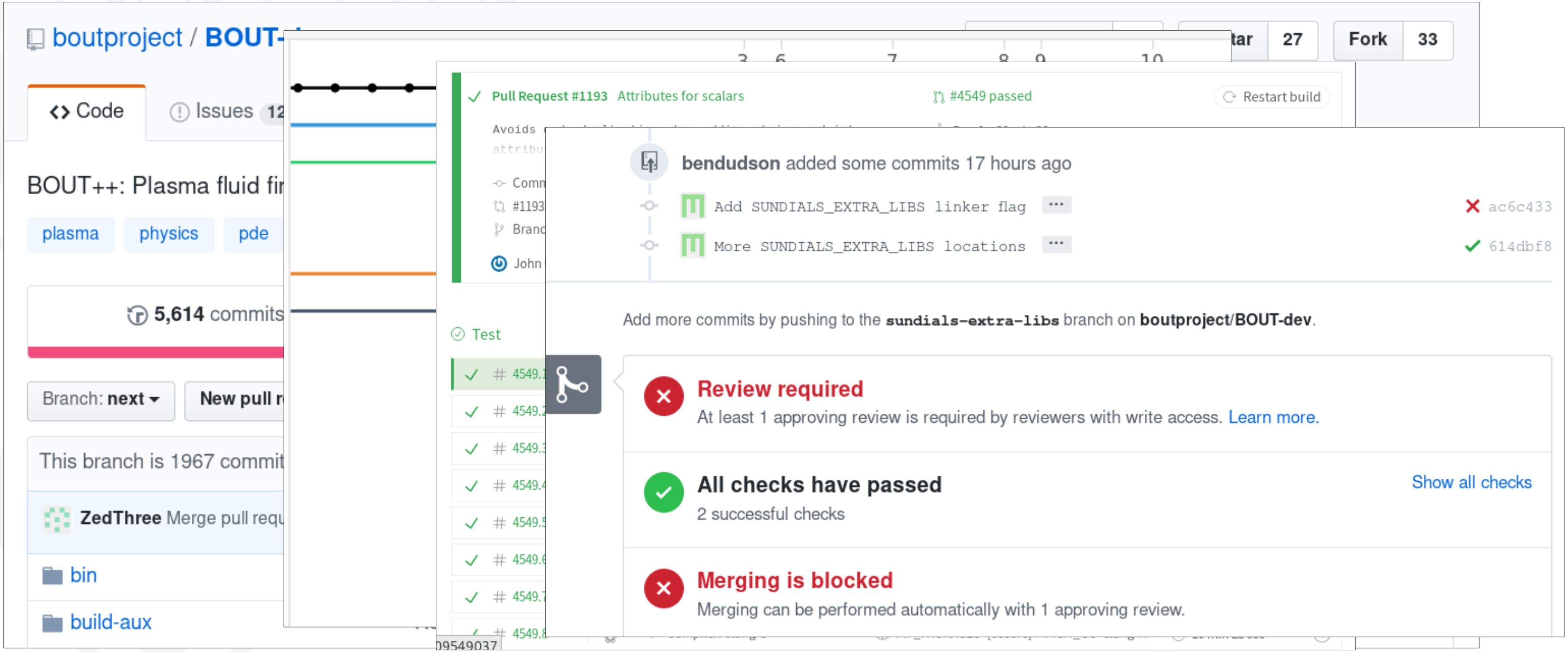
A system of branches with **master**, **next** and features

Development process



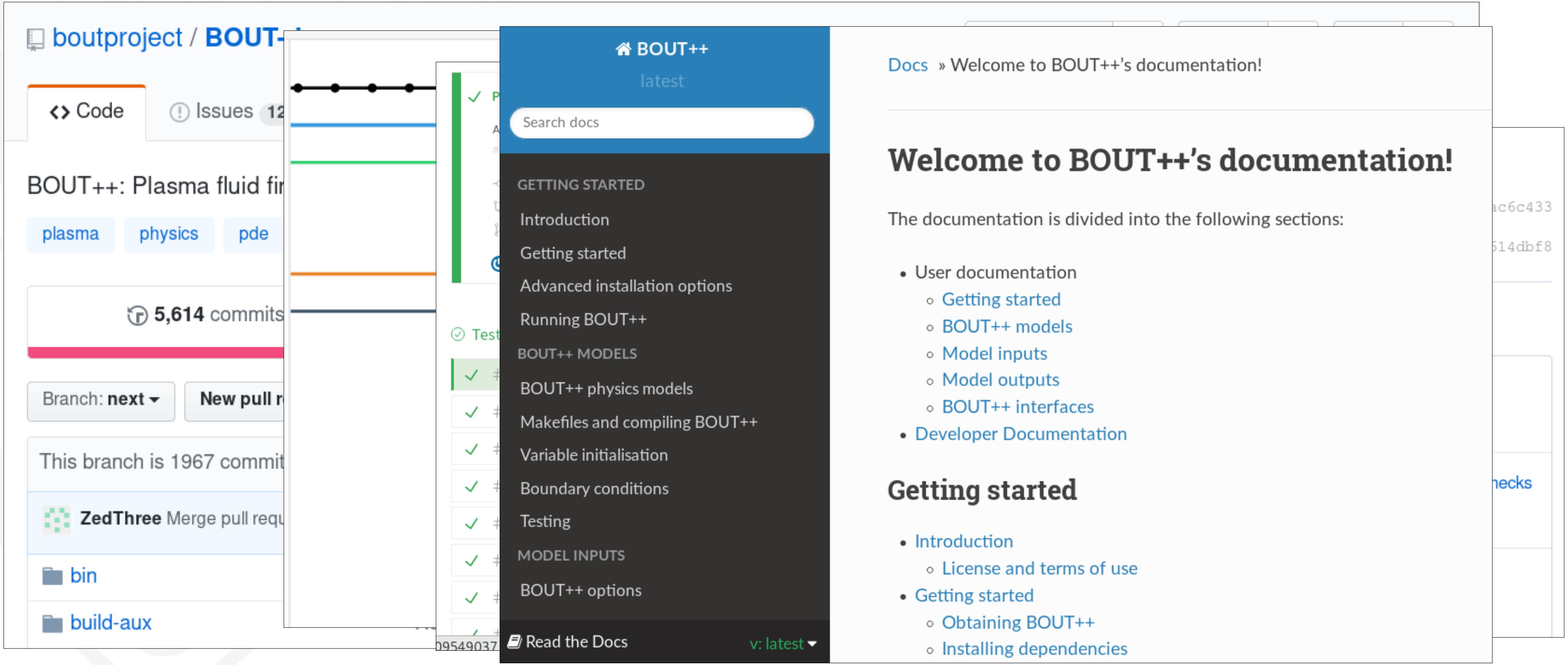
Automated testing using Travis (unit, integrated and MMS tests)

Development process



Merges into next must pass tests and review

Development process



Documentation automatically pulled from Github to Read*the*Docs

Summary

- Fusion plasma physics has many interesting challenges
- BOUT++ has developed as a flexible tool to address a wide range of different problems.
- A good community of users and developers are pushing the code in new directions and adding capabilities
- Development is increasingly professional, with increasing emphasis on maintainability, reproducibility and correctness.

We welcome new contributors and collaborators!